REMARKS

The above Amendments and these Remarks are in reply to the Office Action mailed September 15, 2005. Claims 1-56 were pending in the Application prior to the outstanding Office Action. The present Response amends claims 1, 15, 19, 33, 37, 51, 55, and 56, leaving for the Examiner's present consideration claims 1-56. Reconsideration and withdrawal of the rejections are respectfully requested.

CLAIM REJECTIONS UNDER 35 U.S.C. § 103

1. Claims 1, 3-19, 21-37, 39-56 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,799,718 by *Chan et al.* (hereinafter *Chan*), in view of U.S. Patent No. 6,836,883 by *Abrams et al.* (hereinafter *Abrams*)

To anticipate a claim, every element of the claim must be disclosed within a single reference. The present invention is distinguishable from *Chan* in view of *Abrams* in at least the following ways:

- The present invention teaches a **compilation** framework for a source code written in multiple programming languages, while *Chan* teaches a tool for **development** assistance of such multi-language source code and *Abrams* teaches compilation of multiple source codes each written in a single language.

  More specifically, *Chan* teaches "providing development assistance for multiple languages in an IDE", which "makes the problem of developing such mixed language programs much easier" (column 2, lines 53-60). In other words, it teaches a development tool that helps a programmer to edit/develop multi-language source codes in IDE, not a compilation tool as claimed in the present invention that actually compiles such multi-language source codes into target codes. In fact, none of "syntax highlighting, structure analysis, error reporting, completion assistance and context-sensitive help" (column 2, lines

57-59) is part of a typical compiling process. Although a compiler (572 in Figure 5) is used in *Chan*, it is for the sole purpose of compiling the structural analysis and its errors for display (column 9, line 35-40) and no further details of the compiler is given. Since "structure analysis and error reporting features provide information as the program is developed" (column 1, lines 30-31), compiling such structural analysis and its errors is to help the programmer editing/developing the multi-language source codes, which is totally different from actually compiling such multi-programming-language program into target codes. On the other hand, *Abrams* teaches a compiler in which "different front ends may read different source languages" (column 2, lines 33-34), it is not capable of compiling any source code written in more than one programming languages as the present invention.

- The present invention allows a language module to interact with (invoke) another language module **directly** during the compilation process via a programming interface, while *Chan* does not allow such interaction and *Abrams* only allows interactions **indirectly** via intermediate files.

  In *Chan*, "the code from each language is parsed separately" (column 9 line 17-18) and the modules processing languages L2 (530, 540, 550, etc.) and L3 (534, 542, 556, etc.) do not interact with each other as shown by the flow diagram in Figure 5. Such interaction is not required in *Chan* because it is a code development/editing tool, not a compilation tool as the present invention, so source codes from different languages can be processed in totally separate ways. In contrast, the present invention allows one language module to interact with another (claim 1, 10 and 11), which is especially important for the compilation of source code having nested (multi) programming languages.

  In *Abrams*, "the existing front end complier system" can "be modified so that it can read at least the metadata portion of the intermediate file" and "convert the type and method information in the metadata into the proper form" (col. 12, lines 42-47). In other words,

one language module can interact with another only by reading an intermediate file generated by the other. Since there will always be a necessary time gap between the writing and reading of a file, such interaction between language modules is indirect (through an intermediary) and cannot happen in real time. In contrast, the present invention allows one language module to invoke another module via a language interface, which can be direct programming invocation calls by the "caller" module in real time as required by the compilation of a multi-language source code. This is a faster and more efficient way of interaction between the language modules with no reading or writing of any intermediate files needed.

Therefore, *Chan* in view of *Abrams* cannot anticipate the multi-programming-language compilation process of independent claims 1, 15, 19, 33, 37, 51, 55, and 56. Since claims 3-14 depend claim 1, claims 16-18 depend on claim 15, claims 21-32 depend on claim 19, claim 34-36 depend on claim 33, claims 39-50 depend on claim 37, and claims 52-54 depend on claim 51, *Chan* in view of *Abrams* cannot anticipate claims 1, 3-19, 21-37, and 39-56 for at least this reason, and Applicant respectfully requests that the rejection with respect to these claims be withdrawn.

2. Claims 2, 20 and 38 are rejected under 35 U.S.C. §103(a) as being unpatentable over *Chan* in view of *Abrams* and further in view of *Stone, et al.*, U.S. Patent No. 6,804,686, (hereinafter *Stone*).

Stone teaches "providing a Unified Modeling Language (UML) diagram of a program for display in a graphical user interface" (Abstract). Since it does not teach the compilation of a file written in multiple programming languages, it cannot anticipate the multi-programming-language compilation process in independent claims 1, 19, and 37. As discussed in previous section, these independent claims cannot be anticipated by *Chan* in view of *Abrams* either. Since claim 2 depends on claim 1, claim 20 depends on claim 19 and claim 38 depends on claim 37,

Chan in view of *Abrams* and further in view of *Stone* cannot render claims 2, 20 and 38 obvious under 35 U.S.C. § 103(a) for at least this reason, and Applicant respectfully requests that the rejection with respect to these claims be withdrawn.

CONCLUSION

In light of the above, it is respectfully requested that all outstanding rejections be reconsidered and withdrawn. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this reply, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: 11/15/05 By: _____

David T. Xue
Reg. No. 54,554

FLIESLER MEYER LLP
Four Embarcadero Center, Fourth Floor
San Francisco, California 94111-4156
Telephone: (415) 362-3800
Facsimile: (415) 362-2928
Customer No. 23910